

[首页](#)
[题库](#)
[课程](#)
[排行榜](#)
[求职](#)
[讨论区](#)
[搜索](#) | 
 [APP](#) | 
 [企业版](#) | 
 [登录 / 注册](#)
[首页](#) > [技术交流](#) > 动态规划之0-1背包问题

## 动态规划之0-1背包问题



小小沸沸要加油

编辑于 2016-01-10 10:11:08

回复 4 | 赞 6 | 浏览 4074

## 问题描述：

现有 $n$ 件物品和一个容量为 $c$ 的背包。第 $i$ 件物品的重量是 $w[i]$ ，价值是 $v[i]$ 。已知对于一件物品必须选择取（用1表示）或者不取（用0表示），且每件物品只能被取一次（这就是“0-1”的含义）。求放置哪些物品进背包，可使这些物品的重量总和不超过背包容量，且价值总和最大。

## 求解思路：

假设有5件物品，其重量分别是 $w=[2, 2, 6, 5, 4]$ ，价值分别是 $v=[6, 3, 5, 4, 6]$ ，背包容量为10。在数学问题中这是典型的线性规划问题，我们可以在线性约束范围内求解目标表达式。但是怎么用计算机语言实现呢？我们可以先这样考虑，当背包容量为1时，如何放置物品才能使背包中价值最大；同样当背包容量为2时，如何放置能使背包中价值最大，以此类推，直到背包容量为10。此时我们需要维护一张二维表 $m[i][j]$ ，其中横坐标 $i$ 表示物品，纵坐标表示背包容量（ $1 \leq j \leq 10$ ）。

0-1背包问题的递推二维表

编号	重量	价值	0	1	2	3	4	5	6	7	8	9	10
0	2	6	0	0	6	6	6	6	6	6	6	6	6
1	2	3	0	0	6	6	9	9	9	9	9	9	9
2	6	5	0	0	6	6	9	9	9	9	11	11	14
3	5	4	0	0	6	6	9	9	9	10	11	13	14
4	4	6	0	0	6	6	9	9	12	12	15	15	15

$m[i][j]$ 表示当可以放入前 $i$ 件物品且背包容量为 $j$ 时的最大价值。当只能放入第一件物品即 $i=0$ 时：若背包容量 $j < w[0]$ ，物品不能够被放入背包；若 $j \geq w[0]$ 时，物品可以放入背包，此时 $m[0][j]=v[0]$ 。当可以放入前2件物品即 $i=1$ 时，我们需要进行这样的处理：若 $j < w[1]$ 时，说明第2件物品不能被放入背包内，此时背包的最大价值为背包中只放入第一件物品的最大价值，即 $m[1][j]=m[0][j]$ ；若 $j \geq w[1]$ 时，假设此时背包容量 $j=8$ ，第二件物品可以被放入背包内，那么便会出现两种情况：

(1) 将第二件物品放入背包，那么背包中物品的最大价值是多少呢？因为第二件物品重量为 $w[1]=2$ ，在将第二件物品放入背包之前，背包的容量应为 $j-w[1]=8-2=6$ ，此时背包的最大价值是 $m[0][6]$ ，因此若将第二件物品放入背包，其背包的最大价值 $m[1][j]=m[0][j-w[1]]+v[1]$ ；

(2) 不将第二件物品放入背包，那么此时背包中物品的最大价值依然为只放入第一件物品时背包的最大价值，即 $m[1][j]=m[0][j]$ ；

我们选取 (1) (2) 中价值的较大者作为 $i=1, j=8$ 时背包中的最大价值。

$i=2, 3, 4$ 时的分析同上，直到背包的容量为10，此时 $m[4][10]$ 即为背包中物品的最大价值。

有了上面的分析，我们很容易写出下面的递归关系：

(1)  $i=0$  当 $j < w[0]$ 时， $m[0][j]=0$ ；当 $j \geq w[0]$ 时， $m[0][j]=v[0]$ 。

## 热门推荐



## 技术交流近期热帖

小米

发表于 2016-10-15 11:03:56 回复 (4)

职业发展

发表于 2016-10-24 19:59:07 回复 (1)


 扫一扫下载  
牛客APP

## 牛客网，程序员必备求职神器

QQ群：169195721

微信：www\_nowcoder\_com 关注

微博：牛客网 关注

刷真题、补算法、看面经、得内推



扫一扫 扫码下载牛客APP

[关于我们](#)
[加入我们](#)
[意见反馈](#)
[企业服务](#)
[网站合作](#)
[免责声明](#)
[友情链接](#)

浙ICP备14000860号-2

京公网安备 11011402010078号

牛客网©2016 All Rights Reserved

使用第三方账号直接登录使用吧：

更多

[首页](#) [题库](#) [课程](#) [排行榜](#) [求职](#) [讨论区](#)[搜索](#) | [APP](#) | [企业版](#) | [登录](#) / [注册](#)

$m[i][j]$ ，我们注意到 $m[i][c]$ 表示当背包重量为题目中要求的 $c$ 时背包的最大价值，那么在得到 $m[i][c]$ 之前，我们必然是比较了 $m[i-1][j]-w[i]+v[i]$ 与 $m[i-1][j]$ 的大小，从而决定是否将物品放入背包。所以我们可以利用回溯的方法，若 $m[i][j]=m[i-1][j]$ ，那么物品没有放入背包；否则物品一定被放入背包。因此我们可以从最后一件物品开始，一步一步回退到第一件物品，直到找到所有的物品放入背包的情况。本题中物品的装入情况如表中红色和蓝色部分所示，其中红色表示当前物品被装入背包，蓝色表示没有装入背包。

代码实现：

```
public class Main {
    public static void main(String[] args) {
        int []w={2,2,6,5,4}; //物品重量
        int []v={6,3,5,4,6}; //物品价值
        int c=10; //背包容量
        int []x=new int[5]; //记录物品装入情况，0表示不装入，1表示装入
        x[0]=1; //初始值表示第一个物品已装入背包
        int [][]m=new int[5][c+1]; //需要维护的二维表，为了方便计算加入一列，其中第0列表示
        //背包容量为0时背包的最大价值为0
        /*
        * 初始化第一行，即背包中装入第一件物品
        */
        for(int j=1;j<=c;j++){
            if(j>=w[0]){
                m[0][j]=v[0];
            }
        }
        /*
        * 背包中依次装入其他的物品
        */
        for(int i=1;i<5;i++){
            for(int j=1;j<=c;j++){
                if(j<w[i])m[i][j]=m[i-1][j]; //不装入背包
                else{
                    if(m[i-1][j-w[i]]+v[i]>m[i-1][j]) m[i][j]=m[i-1][j-w[i]]+v[i];
                    //选择价值较大者
                    else m[i][j]=m[i-1][j];
                }
            }
        }
        System.out.println("背包的最大价值为: "+m[w.length-1][c]);
        for(int i=4;i>=1;i--){
            if(m[i][c]>m[i-1][c]){
                x[i]=1; //装入背包
                c-=w[i]; //物品i装入背包之前背包的容量
            }
            else x[i]=0; //没有装入背包
        }
        System.out.print("装入背包的物品编号是: ");
        for(int i=0;i<5;i++){
            if(x[i]==1) System.out.printf("%2d", (i+1));
        }
    }
}
```

[收藏](#) | [分享](#) | [赞\(6\)](#) | [回帖](#)

扫一扫下载  
牛客APP

刷真题、补算法、看面经、得内推

使用第三方账号直接登录使用吧：

[更多](#)

[首页](#)
[题库](#)
[课程](#)
[排行榜](#)
[求职](#)
[讨论区](#)

[搜索](#) | 
 [APP](#) | 
 [企业版](#) | 
 [登录 / 注册](#)

#### 4条回帖

回帖



镜啊哈哈

1#

不错呦!

发表于 2016-01-10 10:18:36

赞(0)

回复(0)



六点的太阳

2#

不错呦!

发表于 2016-01-11 15:23:09

赞(1)

回复(0)



stevenniu

3#

zan

发表于 2016-01-13 21:06:14

赞(0)

回复(0)



牛客883554号

4#

这样不是默认第一个物体加入package里面了吗?把物体的序号颠倒一下就会有bug

发表于 2016-04-10 23:22:34

赞(0)

回复(2)

**牛客2017:** 并不是这样理解的, 二维数组存储的前k个的最大值, 前1个最大值肯定是能包含第一个数据的, 但是前2个或者前n个就不一定包含第一个数据了, 你可以实现一下代码, 当把输入第一行输入5 4和2 6颠倒数据后(5 4放在第一行), 还是能得出正确的输出15的。

2016-04-11 02:05:53

赞(0)

回复

**蜡笔小树:** 楼主代码后面部分, 就是选出放入背包物品部分, 对于最后一个物品选择, 就是i=0时候, 应该判断一下(当前i = 0, c = 4) m[i][c] > 0吧, 大于零才放入, 等于0就不放入吧, 不然确实会有这位同学提出的这个问题。

2016-08-05 06:58:37

赞(0)

回复

请输入你的观点

回复



扫一扫下载  
牛客APP

刷真题、补算法、看面经、得内推

使用第三方账号直接登录使用吧:

更多

[首页](#) [题库](#) [课程](#) [排行榜](#) [求职](#) [讨论区](#)

[🔍 搜索](#) | [📱 APP](#) | [💼 企业版](#) | [登录 / 注册](#)

在这里畅所欲言你的看法吧!

回帖



扫一扫下载  
牛客APP

刷真题、补算法、看面经、得内推

使用第三方账号直接登录使用吧:

更多